4A

УДК 004.93 004.89

***А.В. Агарков***
Институт проблем искусственного интеллекта МОН Украины и НАН Украины,
г. Донецк, Украина
aav@iai.donetsk.ua

# Организация словаря на основе применения метода построения дополнительного графа-пирамиды

Предлагается метод организации словаря на основе использования графа-пирамиды, который позволяет без увеличения размера словаря учитывать случаи пропуска, замены и наличия лишних букв в распознаваемых словах. Для хранения совпадающих фрагментов слов используется только одна вершина пирамиды-словаря, что позволяет определять общие фрагменты слов без проведения дополнительного поиска.

## Введение

При автоматическом распознавании текстов и речи зачастую возникает проблема соотнесения распознанного слова со словом из существующего словаря (далее – термином). Это позволяет повысить точность распознавания даже в случае неправильного или неоднозначного распознавания отдельных букв (фонем, визем). Для быстрого поиска в словаре соответствующего термина применяются различные методы, основанные на специальной организации словарей – бинарные деревья, красно-чёрные деревья, слоёные списки [1]. В случае если предполагается, что соответствие между распознаваемым словом и словом из словаря может быть неполным (пропуск, замена, наличие лишних букв), то используются метод расширения выборки, trie-деревья [2], [3], метод n-грамм и др. Для того, чтобы учесть возможность отсутствия начальных и конечных букв в слове, словарь расширяют за счёт включения всех суффиксов и префиксов.

**В работе предлагается метод** организации словаря на основе использования пирамиды, метод построения которой представлен в работах [4], [5]. Данный метод, в отличие от существующих аналогов, позволяет без увеличения размера словаря учитывать случаи пропуска, замены и наличия лишних букв в распознаваемых словах.

## Линейная пирамида

В работах [4], [5] даны общие свойства пирамид и рассмотрены несколько их частных случаев. В данной работе используется только один тип пирамид, наиболее подходящий для организации словаря из терминов, которые представляют собой конечные последовательности символов алфавита. Данный тип пирамид получил название линейных.

Линейной пирамидой на графе $G$ называется граф $P$, имеющий следующие свойства:

– каждая вершина пирамиды $v_{i,j}$ принадлежит определённому уровню $i$;

– уровни пирамиды упорядочены от нижнего (первого) до верхнего (последнего) ($i = 1,...,N$);

– каждой вершине $v_{i,j}$ пирамиды соответствует пара смежных вершин $p(v_{i,j})$, принадлежащих предыдущему уровню (кроме вершин графа *G*, который рассматривается как самый нижний уровень пирамиды *P*);

– каждая из вершин, принадлежащая $p(v_{i,j})$, является родителем для вершины $v_{i,j}$, которая, в свою очередь, является дочерней для вершин множества $p(v_{i,j})$;

– вершины одного уровня соединяются ребром только в том случае, если они имеют общего родителя;

– вершины одного уровня $v_{i,j}$ и $v_{i,k}$ соединяются ребром только в том случае, если общие родители пар вершин $p(v_{i,j})$ и $p(v_{i,k})$ не совпадают;

– вершина пирамиды *P*, не имеющая дочерних вершин, называется верхушкой пирамиды.

Подграф $p(v_{i,j})$ называется родительским для вершины $v_{i,j}$.

Каждая вершина пирамиды имеет маркер, который зависит только от маркеров родительских вершин и ребер, соединяющих родителей. То есть по маркеру вершины можно определить маркеры её предков, и наоборот – по маркерам предков можно определить маркер их потомка (потомков).

Каждой вершине пирамиды *P* соответствует подграф графа *G* который в случае линейной пирамиды представляет собой цепь. Если маркеры двух вершин пирамиды одного уровня совпадают, то подграфы, им соответствующие, изоморфны.

Пирамида образуется в результате построения, то есть начиная с основного графа (нулевого уровня) проводится следующая процедура – граф, соответствующий одному уровню, разбивается на подграфы, для каждого из которых в следующем уровне образуется новая вершина. Эти новые вершины являются дочерними для вершин соответствующих подграфов предыдущего уровня, которые, в свою очередь, являются родителями для нововозведенных вершин.

На рис. 1 представлен пример линейной пирамиды, построенной на графе, состоящем из вершин a, b, c, d, который представляет собой цепь. Пунктирными линиями обозначены уровни пирамиды. В данной пирамиде три уровня (нулевой уровень – исходный граф). Если рассмотреть отдельную вершину $v_{2,1}$, принадлежащую второму уровню, то для неё родителями являются вершины $v_{1,1}$ и $v_{1,2}$, принадлежащие первому уровню. Дочерней для данной вершины является $v_{3,1}$. Данная вершина имеет с $v_{2,2}$ общего родителя $v_{1,2}$, поэтому $v_{2,1}$ и $v_{2,2}$ соединены ребром. Подграф основного графа, который соответствует $v_{2,1}$, образован вершинами a, b, c.

Вершина $v_{3,1}$ не имеет дочерних вершин, поэтому является верхушкой данной пирамиды, и ей соответствует весь основной граф. Она является потомком для всех остальных вершин этой пирамиды, которые, в свою очередь, являются предками для верхушки.

Таким образом, каждый уровень пирамиды представляет собой граф, каждой вершине которого соответствует подграф предыдущего уровня. Вершины данного подграфа соединяются рёбрами, если цепи, которые им соответствуют, различаются только парой крайних вершин.

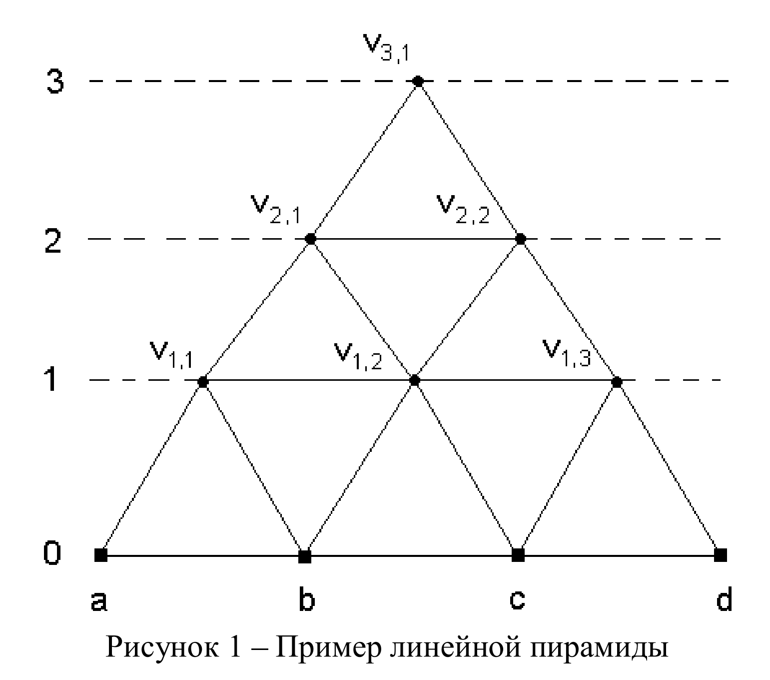



Рисунок 1 – Пример линейной пирамиды

## Использование графа для представления слов и словаря

Каждое слово может быть представлено в виде графа, каждой вершине которого соответствует отдельная буква. А дуги (ориентированные рёбра) определяют последовательность букв. Данный граф является цепью. Однако если в слове есть одинаковые буквы, то различным вершинам данного графа будет соответствовать одна буква. Для того чтобы этого избежать, можно использовать граф, каждой вершине которого соответствует одна буква и не существует буквы, которой бы соответствовало более одной вершины. Дуги определяют последовательность букв в слове. При этом если в слове есть одинаковые буквы, то в цепь, определяющую слово, входит цикл. Если в слове подряд идут несколько одинаковых букв, то существует петля, инцидентная соответствующей вершине.

Таким образом, всякий словарь может быть представлен в виде графа, каждой вершине которого соответствует буква алфавита (причём не существует двух вершин, которым бы соответствовала одна и та же буква), дуги, соединяющие вершины, определяют последовательность букв в словах. Каждое слово из данного словаря определяется цепью в данном графе. По сути, данный граф является мультиграфом. Количество вершин в данном графе равно количеству букв в алфавите, а максимальное количество рёбер – его квадрату.

Для того, чтобы обозначить цепи, соответствующие словам, используется линейная пирамида.

Каждая вершина линейной пирамиды имеет ровно двух родителей, которые принадлежат одному уровню, являются смежными и, в свою очередь, имеют только одного общего родителя. Подграфами разбиения для построения линейной пирамиды являются пары смежных вершин и инцидентная им дуга. Точнее сказать, каждому ребру (и соответственно инцидентным данному ребру вершинам) соответствует подграф разбиения и вершина более высокого уровня пирамиды. Это относится и к петлям.

Таким образом, вершинам данной пирамиды соответствуют слова из словаря и их фрагменты. Если какое-то слово совпадает с фрагментом другого слова или какие-либо фрагменты слов совпадают, то им соответствует одна вершина данной пирамиды.

Множество дочерних вершин каждой вершины пирамиды делится на правое и левое подмножества (считается, что буквы в слове следуют слева направо).

# Построение пирамиды-словаря

Каждому слову соответствует цепь в графе. Построение соответствующего участка пирамиды проводится в результате следующей процедуры – на цепи, которая соответствует начальному фрагменту слова, строится линейная пирамида, затем к концу цепи добавляется ещё одна вершина, соответствующая следующей букве в слове, и пирамида достраивается. Так продолжается до тех пор, пока не закончится цепь – слово. В начале процедуры цепь инициализируется дугой и инцидентными ей вершинами, которые соответствуют первым двум буквам слова.

То есть процедура построения участка линейной пирамиды для слова, которому соответствует цепь $C=(v_1,...,v_n)$, $n$ – количество букв в слове, выглядит следующим образом:

$C_P=(v_1,v_2)$ – инициализация текущей цепи.
Построить линейную пирамиду $LP(C_P)$ на $C_P$;
Цикл от $i=2$ до $i=n$
{
$C_N=C_P+v_i=(v_1,...,v_i)$;
достроить пирамиду $LP(C_P)$ до пирамиды $LP(C_N)$.
$C_P=C_N$;
}

Общий алгоритм построения пирамиды-словаря приведен ниже.

Обозначим $A=\{a_1,a_2,...,a_{NA}\}$ – алфавит – множество букв, $NA$ – количество букв;

$w=(a_{w_1},a_{w_2},...,a_{w_{wn}})$ слово, последовательность букв из алфавита $A$, причём буквы могут повторяться, $wn$ – количество букв в слове $w$; $D=\{w_1,w_2,...,w_{DN})$ – словарь, множество слов, $DN$ – количество слов в словаре; $G=(V,E)$ – граф алфавита.

Свойства графа алфавита $G=(V,E)$:

– $V=\{v_1,v_2,...v_{NA}\}:\exists f:f(a_i\in A)=v_i, f^{-1}(v_i\in V)=a_i\in A$, то есть каждой букве из алфавита $A$ соответствует одна вершина графа $G$;

– $E=\{d(v_i,v_j):\exists w\in D, k<wn:(a_{w_k},a_{w_{k+1}})\in w, f(a_{w_k})=v_i, f(a_{w_{k+1}})=v_j\}$, $d(v_i,v_j)$ – дуга, инцидентная вершинам $v_i$ и $v_j$, то есть любым двум соседним буквам любого слова, принадлежащего словарю $D$, соответствует дуга графа $G$;

– если $\exists d=d(v_i,v_j):v_i=v_j\Rightarrow d(v_i,v_j)$ – петля.

Обозначим $P=(V^P,E^P)$ – линейная пирамида-словарь. Свойства $P=(V^P,E^P)$:

– $G\subset P$;
$\forall v^P\in V^P\setminus V\,\exists v_i^P\in V^P, v_j^P\in V^P, e^P\in E^P$:

– $v_i^P=Lp(v^P)$ – *левый родитель*, $v_i^P=Rp(v^P)$ – *правый родитель*,
$e^P=d(v_i^P,v_j^P)\Rightarrow vL(e^P)=v_i^P$, $vR(e^P)=v_j^P$

– $\forall v^P \in V^P \, \exists Ls(v^P) = \{v_i^P \in V^P : v^P = Rp(v_i^P)\}$ ($Ls(v^P)$ – множество левых дочерних вершин $v^P$);

– $\forall v^P \in V^P \, \exists Rs(v^P) = \{v_i^P \in V^P : v^P = Lp(v_i^P)\}$ ($Rs(v^P)$ – множество левых дочерних вершин $v^P$);

– $\exists Se(e^P) = v^P : \forall e^P \in E^P \, \exists v^P \in V^P \setminus V : e^P = d(Lp(v^P), Rp(v^P))$;

– $mrk(v^P)$ – маркер вершины $v^P$.

Обозначим $M_w(v^P)$ – множество слов, которым принадлежит фрагмент, соответствующий вершине $v^P$.

Алгоритм построения пирамиды-словаря основан на использовании рекурсивной функции для создания новых элементов на основе данной дуги. Обозначим данную функцию ARC_PROC($d$, $w$). Алгоритм функции ARC_PROC($d$, $w$) имеет следующий вид:

ARC_PROC($d$, $w$)
{
  $v_L^P$ – текущая левая вершина;
  $v_R^P$ – текущая правая вершина;
  $d_p$ – текущая дуга;
  если $\neg\exists v^P = Se(d)$ то
  {
    создать вершину $v^P = Se(d)$;
    $v_R^P = v^P$;
    $Rs(vL(d)) = Rs(vL(d)) \cup v^P$;
    $Ls(vR(d)) = Ls(vR(d)) \cup v^P$;
    $M_w(v^P) = M_w(v^P) \cup \{w\}$;
  }
  иначе $v_R^P = Se(d)$;
  $mrk(vL(d)) = 0$;
  $mrk(vR(d)) = indx(w)$;
  $mrk(v_R^P) = indx(w)$;

  если $\exists v_i^P \in Ls(vL(d)) : mrk(v_i^P) = indx(w)$ то
  {
  $v_L^P = v_i^P$;
  если $\neg\exists d = d(v_L^P, v_R^P)$ то
  {
    создать $d = d(v_L^P, v_R^P)$;
    $d_p = d$;
    }
    иначе $d_p = d(v_L^P, v_R^P)$;
    ARC_PROC($d_p$, $w$)
  }
}.

Алгоритм построения пирамиды-словаря имеет следующий вид:

$G = (V, E = \varnothing)$;

for ( $i = 1; i \le DN$ )

{

$w^i$ – $i$-ое слово из словаря $D$;

for ( $j = 1;\ j \le wn^i$ )

{

если $\exists d = d(f(a_{w^i_j}), f(a_{w^i_{j+1}}))$ то $d_p = d(f(a_{w^i_j}), f(a_{w^i_{j+1}}))$;

иначе

{

создать дугу $d = d(f(a_{w^i_j}), f(a_{w^i_{j+1}}))$;

$d_p = d(f(a_{w^i_j}, a_{w^i_{j+1}}))$;

}

ARC_PROC( $d_p$ , $w^i$ );

}

}

На рис. 2 представлен фрагмент пирамиды-словаря, соответствующий слову «мама». Для наглядности вершины пирамиды маркированы фрагментами термина, которым они соответствуют. Пунктирной линией соединены вершина и дуга, которая является основой для соответствующего подграфа разбиения.

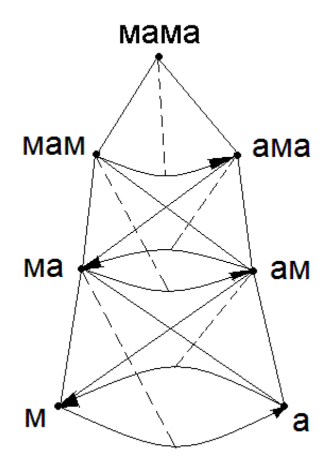



Рисунок 2 – Пример фрагмента пирамиды-словаря, соответствующий слову «мама»

# Поиск по словарю на основе построения пирамиды поиска

В качестве входных данных выступает последовательность множеств $S = (L_1, L_2, ..., L_{NL})$, каждое из которых состоит из вариантов букв – $L_i = \{a^i_1, a^i_2, ..., a^i_{nl_i}\}$. Необходимо из каждого множества выбрать те буквы, последовательность которых составляла бы слова из словаря и(или) их фрагменты.

То есть необходимо найти такие последовательности $w^S_i = (a^k_{m^i_0}, a^{k+1}_{m^i_1}, ..., a^{k+n_i}_{m^i_{n_i}})$, где $k \ge 1, (k + n_i) \le NL$, которые бы соответствовали словам из словаря и их фрагментам.

После чего выполняется поиск наиболее подходящей условию задачи последовательности $w_i^S$, что является отдельной задачей и здесь рассматриваться не будет.

Поиск множества $\{w_i^S\}$ осуществляется с помощью построения пирамиды поиска по образу и подобию пирамиды-словаря.

Обозначим $P^S = (V^S, E^S)$ – пирамиду поиска. Свойства $P^S = (V^S, E^S)$:

– $\exists sv(v^S \in V^S): \forall v^S \in V^S \exists v^P \in V^P : sv(v^S) = v^P$;

– $\exists se(e^S \in E^S): \forall e^S \in E^S \exists e^P \in E^P : se(e^S) = e^P$;

– $V_G^S = \{v^S \in V^S : sv(v^S) \in G\}$;
$\forall v^S \in V^S \setminus V_G^S \exists v_i^S \in V^S, v_j^S \in V^S, e^S \in E^S$;

– $v_i^S = Lp(v^S)$ – левый родитель, $v_i^S = Rp(v^S)$ – правый родитель, ;
$e^S = d(v_i^S, v_j^S) \Rightarrow vL(e^S) = v_i^S,\ vR(e^S) = v_j^S$;

– $\forall v^S \in V^S \exists Rs(v^S) = \{v_i^S \in V^S : v^S = Lp(v_i^S)\}$ ($Rs(v^S)$ – множество левых дочерних вершин $v^S$);

– $\forall v^S \in V^S \exists Ls(v^S) = \{v_i^S \in V^S : v^S = Rp(v_i^S)\}$ ($Ls(v^S)$ – множество левых дочерних вершин $v^S$);

– $\exists Se(e^S) = v^S : \forall e^S \in E^S \exists v^S \in V^S \setminus V : e^S = d(Lp(v^S), Rp(v^S))$;

– $\forall v^S, v_l^S, v_r^S, e^S : v_l^S = Lp(v^S), v_r^S = Rp(v^S), Se(e^S) = v^S \Leftrightarrow$
$v^P = sv(v^S), v_l^P = sv(v_l^S), v_r^P = sv(v_r^S), e^P = se(e^S): v_l^P = Lp(v^P), v_r^P = Rp(v^P), Se(e^P) = v^P$;

– $\forall v^S, v_l^S, v_r^S : v_l^S \in Ls(v^S), v_r^S \in Rs(v^S), \Leftrightarrow$
$v^P = sv(v^S), v_l^P = sv(v_l^S), v_r^P = sv(v_r^S): v_l^P = Ls(v^P), v_r^P = Rs(v^P).$

Каждой вершине $v^S$ пирамиды $P^S$ соответствует цепь вершин, принадлежащих множеству $V_G^S$ $\forall v_i^S \in V^S \setminus V_G^S \exists C(v_i^S) = (v_{m_1^i}^S, ..., v_{m_{n_i}^i}^S : v_{m_j^i}^S \in V_G^S)$, причём последовательность букв $(f(sv(v_{m_1^i}^S)), ..., f(sv(v_{m_{n_i}^i}^S))$ соответствует слову из словаря и(или) фрагменту слова, которое соответствует вершине $sv(v_i^S) \in V^P \setminus V$.

Алгоритм построения пирамиды поиска основан на использовании рекурсивной функции SARC_PROC($d$), которая на основе данной дуги создаёт новые элементы пирамиды. Алгоритм SARC_PROC($d$) имеет следующий вид:

```
SARC_PROC(d)
{
  v_L^P – текущая левая вершина;
  v_R^P – текущая правая вершина;
  d_p – текущая дуга;
        если ¬∃v^P = Se(d) то
{
        создать вершину v^P = Se(d);
        v_R^P = v^P;
        Rs(vL(d)) = Rs(vL(d)) ∪ v^P;
```

$Ls(vR(d)) = Ls(vR(d)) \cup v^P$ ;
}
иначе $v_R^P = Se(d)$ ;

если $\exists v_i^P \in Ls(vL(d)) : \exists d(sv(v_i^P), sv(v_R^P)) \in E^P$ то
{
$v_L^P = v_i^P$ ;
если $\neg\exists d = d(v_L^P, v_R^P)$ то
{
создать $d = d(v_L^P, v_R^P)$ ;
$d_p = d$ ;
}
иначе $d_p = d(v_L^P, v_R^P)$ ;
SARC_PROC( $d_p$ )
}
}

В алгоритме построения пирамиды поиска используются следующие вспомогательные множества и переменные – $G_1, G_2$ – вспомогательные множества вершин графа $V_G^S$ ($G_i = \{v_k^i\}$), $nG_1, nG_2$ – количество элементов во множествах $G_1, G_2$. Алгоритм построения пирамиды поиска имеет следующий вид:

$G_1 = \varnothing$ ;
$G_2 = \varnothing$
for ( $k = 1; k \le NL$ )
{
for ( $i = 1; i \le nL_k$ )
{
создать вершину $v^S \in V_G^S \subset V^S : f(sv(v^S)) = a_i^k$ ;
$G_2 = G_2 \cup \{v^S\}$ ;
}

for ( $i = 1; i \le nG_1$ )
{
for ( $j = 1; j \le nG_2$ )
{
если $\exists e^P = d(sv(v_i^1), sv(v_j^2)) \in G$ то
{
создать дугу $d_p = d(v_i^1, v_j^2)$ ;
SARC_PROC( $d_p$ );
}
}
}
$G_1 = G_2$ ;
$G_2 = \varnothing$ ;
}

Каждой вершине пирамиды $P^S$, не имеющей дочерних вершин, будет соответствовать полное слово из словаря или(и) фрагмент слова из словаря. Если в искомом слове буквы пропущены или заменены на другие – необходимо проанализировать все найденные фрагменты на предмет возможности составления из них слов с пропусками букв, что алгоритмически труда не представляет.

# Заключение

Предложенный способ организации словаря в виде пирамиды позволяет использовать для хранения совпадающих фрагментов различных слов только одну вершину. Это позволяет определять общие фрагменты слов из словаря без проведения дополнительного поиска, а также кодировать их номером соответствующей вершины. Данный метод организации словаря в совокупности с поиском, основанным на методе построения пирамиды, позволяет учитывать случаи отсутствующих, неправильных и лишних букв в распознаваемых словах. Также данный подход позволяет рассматривать различные варианты слов в случае неоднозначного введения отдельных букв.

# Литература

1. Озкарахан Э. Машины баз данных и управление базами данных / Озкарахан Э. – М. : Мир, 1989.
2. Shang H. Tries for Approximate String Matching / H. Shang, T.H. Merrett // IEEE Trans. on Knowledge and Data Engineering – special issue on Digital Libraries / ed. Nabil R. Adam. – 1996. – P. 540-547.
3. Merrett T.H. Database Structures, Based on Tries, for Text, Spatial, and General Data / T.H. Merrett, H. Shang, X. Zhao // International Symposium on Cooperative Database Systems for Advanced Applications (Kyoto, Dec. 5-7, 1996). – Kyoto, 1996. – P. 316-324.
4. Агарков А.В. Метод сравнения двух графов за полиномиальное время / А.В. Агарков // Искусственный интеллект. – 2003. – № 4. – С. 172-184.
5. Агарков А.В. Поиск изоморфных пересечений двух графов за полиномиальное время / А.В. Агарков // Искусственный интеллект. – 2007. – № 2. – С. 62-74.

***А.В. Агарков***
**Організація словника на основі застосування методу побудови додаткового графа-піраміди**
Пропонується метод організації словника на основі використання графа-піраміди, що дозволяє без збільшення розміру словника враховувати випадки пропуску, заміни і наявності зайвих букв у словах, що розпізнаються. Для збереження збіжних фрагментів слів використовується тільки одна вершина піраміди-словника, що дозволяє визначати загальні фрагменти слів без проведення додаткового пошуку.

***A.V. Agarkov***
**Dictionary Construction on the Basis of Application of Additional Graph-Pyramid Building Method**
It is proposed the dictionary construction method on the basis of the graph-pyramid using which allows to consider without increase in the size of the dictionary cases omission, replacement and presence of superfluous letters in recognized words. For storage of conterminous fragments of words only one pyramid-dictionary vertex is used which allows to define the common word fragments without carrying out of additional search.

*Статья поступила в редакцию 15.06.2009.*